

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method for finding ~~an~~ a global extrema for a plurality of values stored in an n-dimensional array having a plurality of processing elements, the method comprising:

determining within each of said processing elements a first dimensional extrema for a first dimension of said n-dimensional array, wherein said first dimensional extrema is related to one or more local extrema of a plurality of values stored in said processing elements in said first dimension and wherein said first dimensional extrema has a most significant byte and a least significant byte;

determining within each of said processing elements a next dimensional extrema for a next dimension of said n-dimensional array, wherein said next dimensional extrema is related to one or more of said first dimensional extrema and wherein said next dimensional extrema has a most significant byte and a least significant byte; ~~and~~

repeating said determining within each of said processing elements a next dimensional extrema for each of said n-dimensions, wherein each of said next dimensional extrema is related to a dimensional extrema from a previously selected dimension, until the global extrema is found; and;

saving said global extrema.

2. (original) The method of claim 1 wherein said determining within each of said processing elements a first dimensional extrema for a first dimension of said n-dimensional array comprises:

receiving a set of local extrema from one or more of said processing elements within said first dimension;

separating said set of local extrema into an odd numbered set of local extremas and an even numbered set of local extremas;

separating each of said odd numbered local extrema into at least one of an odd most significant byte and an odd least significant byte;

separating each of said even numbered local extrema into at least one of an even most significant byte and an even least significant byte;

- determining an odd extrema from said odd numbered set;
 - determining an even extrema from said even numbered set; and
 - determining said first dimensional extrema for a first dimension from said odd extrema and said even extrema.
3. (original) The method of claim 2 wherein said receiving a set of local extrema from one or more of said processing elements within said first dimension comprises:
- receiving a burst of said odd and even least significant bytes; and
 - receiving a burst of said odd and even most significant bytes.
4. (original) The method of claim 3 further comprising:
- selecting a burst length for said burst of odd and even least significant bytes and a burst length for said odd and even most significant bytes to minimize the amount of lost cycles within said processing elements.
5. (original) The method of claim 1 wherein said determining within each of said processing elements a next dimensional extrema for a next dimension of said n-dimensional array comprises:
- receiving a set of said first dimensional extrema from one or more of said processing elements within said next dimension;
 - separating said set of first dimensional extrema into an odd numbered set and an even numbered set;
 - separating each of said odd numbered set into at least one of an odd most significant byte and an odd least significant byte;
 - separating each of said even numbered set into at least one of an even most significant byte and an even least significant byte;
 - determining a odd extrema from said odd numbered set;
 - determining an even extrema from said even numbered set; and
 - determining said next dimensional extrema for a next dimension from said odd extrema and said even extrema.
6. (original) The method of claim 5 wherein said receiving a set of said first dimensional extrema from one or more of said processing elements within said next dimension comprises:
- receiving a burst of said odd and even least significant bytes; and

receiving a burst of said odd and even most significant bytes.

7. (original) The method of claim 6 further comprising:
 - selecting a burst length for said burst of odd and even least significant bytes and a burst length for said odd and even most significant bytes to minimize the amount of lost cycles within said processing elements.
8. (original) The method of claim 1 wherein said repeating said determining within each of said processing elements a next dimensional extrema for each of n-said dimensions comprises:
 - receiving a set of dimensional extrema from a previously selected dimension from one or more of said processing elements within a currently selected dimension;
 - separating said set of dimensional extrema from said previously selected dimension into an odd numbered set and an even numbered set;
 - separating each of said odd numbered set into at least one of an odd most significant byte and an odd least significant byte;
 - separating each of said even numbered set into at least one of an even most significant byte and an even least significant byte;
 - determining a odd extrema from said odd numbered set;
 - determining an even extrema from said even numbered set; and
 - determining said next dimensional extrema for said next dimension from said odd extrema and said even extrema.
9. (original) The method of claim 2 wherein determining an odd extrema from said odd numbered set comprises:
 - loading the least significant byte of an odd numbered local extrema into a least significant odd byte register;
 - loading the most significant byte of said odd numbered local extrema into a most significant odd byte register;
 - comparing the contents of said least significant odd byte register to the least significant byte of another odd numbered local extrema and setting a carry flag relative to said comparison;
 - comparing the contents of said most significant odd byte register to the most significant byte of said another odd numbered local extrema and setting an odd flag relative to said comparison;
 - and

conditionally updating said most significant odd byte register and said least significant odd byte register relative to said odd flag.

10. (original) The method of claim 2 wherein said determining an even extrema from said even numbered set comprises.

loading the least significant byte of an even numbered local extrema into a least significant even byte register;

loading the most significant byte of said even numbered local extrema into a most significant even byte register;

comparing the contents of said least significant even byte register to the least significant byte of another even numbered local extrema and setting a carry flag relative to said comparison;

comparing the contents of said most significant even byte register to the most significant byte of said another even numbered local extrema and setting an even flag relative to said comparison;
and

conditionally updating said most significant even byte register and said least significant even byte register relative to said even flag.

11. (original) The method of claim 9 further comprising repeating said comparing the contents of said most significant odd byte register, said comparing the contents of said least significant odd byte register, and said conditionally updating said most significant even byte register and said least significant even byte register for each of said odd numbered local extrema within said set.

12. (original) The method of claim 10 further comprising repeating said comparing the contents of said most significant even byte register, said comparing the contents of said least significant even byte register, and said conditionally updating said most significant even byte register and said least significant even byte register for each of said even numbered local extrema within said set.

13. (original) The method of claim 2 wherein said determining said next dimensional extrema for a next dimension from said odd extrema and said even extrema further comprises:

loading the least significant byte of said odd extrema into a least significant odd byte register and the most significant byte of said odd extrema into a most significant odd byte register;

loading the least significant byte of said even extrema into a least significant even byte register and the most significant byte of said even extrema into a most significant even byte register;

comparing the contents of said least significant even byte register to the contents of said least significant odd byte register and setting a carry flag relative to a result of said comparison;

comparing the contents of said most significant even byte register to the contents of said most significant odd byte register and setting an extrema flag relative to a result of said comparison; and

conditionally updating said most significant even byte register and said least significant even byte register relative to said extrema flag.

14. (currently amended) A method comprising:

identifying extrema within a data stream as having one of an odd or an even position, said extrema having a most significant byte and a least significant byte;

processing said extrema having an odd position to produce an odd extrema, said odd extrema having a most significant byte and a least significant byte;

processing said extrema having an even position to produce an even extrema, said even extrema having a most significant byte and a least significant byte; ~~and~~

determining a dimensional extrema from said odd extrema and said even extrema, said dimensional extrema having a most significant byte and a least significant byte; and

saving said dimensional extrema.

15. (original) The method of claim 14 wherein said processing said extrema having an odd position comprises:

loading the least significant byte of an odd numbered local extrema into a least significant odd byte register;

loading the most significant byte of said odd numbered local extrema into a most significant odd byte register;

comparing the contents of said least significant odd byte register to the least significant byte of another odd numbered local extrema and setting a carry flag relative to said comparison;

comparing the contents of said most significant odd byte register to the most significant byte of said another odd numbered local extrema and setting an odd flag relative to said comparison; and

conditionally updating said most significant odd byte register and said least significant odd byte register relative to said odd flag.

16. (original) The method of claim 14 wherein said identifying extrema within a data stream as having one of an odd or an even position comprises:

loading a burst of said least significant bytes of said odd and said even positioned extrema, wherein said odd positioned least significant bytes are loaded into a first plurality of registers and said even positioned least significant bytes are loaded into a second plurality of registers; and

loading a burst of said most significant bytes of said odd and said even positioned extrema, wherein said odd positioned most significant bytes are loaded into said first plurality of registers and said even positioned most significant bytes are loaded into said second plurality of registers.

17. (original) The method of claim 14 wherein said processing said extrema having an even position comprises:

loading the least significant byte of an even numbered local extrema into a least significant even byte register;

loading the most significant byte of said even numbered local extrema into a most significant even byte register;

comparing the contents of said least significant even byte register to the least significant byte of another even numbered local extrema and setting a carry flag relative to said comparison;

comparing the contents of said most significant even byte register to the most significant byte of said another even numbered local extrema and setting an even flag relative to said comparison; and

conditionally updating said most significant even byte register and said least significant even byte register relative to said even flag.

18. (original) The method of claim 15 further comprising repeating said comparing the contents of said most significant odd byte register, said comparing the contents of said least significant odd byte register, and said conditionally updating said most significant even byte register and said least significant even byte register for each of said odd numbered local extrema within said set.

19. (currently amended) The method of claim ~~16~~ 17 further comprising repeating said comparing the contents of said most significant even byte register, said comparing the contents of said least significant

even byte register, and said conditionally updating said most significant even byte register and said least significant even byte register for each of said even numbered local extrema within said set.

20. (original) The method of claim 14 further comprising:

determining a next dimensional extrema for a next dimension for a n-dimensional array, wherein said next dimensional extrema is related to said dimensional extrema, said next dimensional extrema having a most significant byte and a least significant byte.; and
repeating said determining a next dimensional extrema for each of said n-dimensions, wherein each of said next dimensional extrema is related to a dimensional extrema from a previously selected dimension.

21. (currently amended) A method for determining a dimensional extrema for a plurality of values stored in an n-dimensional array of processing elements, comprising:

loading odd numbered extrema from odd positions within a set of said processing elements in a first dimension into a first plurality of registers;

loading even numbered extrema from even positions within a set of set processing elements into a second plurality of registers;

comparing certain of said loaded odd numbered extrema to produce an odd extrema, said odd extrema having a most significant byte and a least significant byte;

comparing certain of said loaded even numbered extrema to produce an even extrema, said even extrema having a most significant byte and a least significant byte; ~~and~~

producing a dimensional extrema in response to said odd extrema and said even extrema, said dimensional extrema having a most significant byte and a least significant byte; and
saving said dimensional extrema.

22. (original) The method of claim 21 wherein said loading odd numbered extrema from a set of said processing elements in a first dimension into a first plurality of registers comprises:

loading said least significant byte of an extrema having an odd position into a first register;
transferring said least significant byte of said extrema in said first register into a second register and loading said most significant byte of said extrema into said first register;

transferring said most significant byte of said extrema in said first register into a third register and loading said least significant byte of another extrema having an odd position into said first register; and

transferring said least significant byte of said another extrema in said first register into a fourth register and loading said most significant byte of said another extrema into said first register.

23. (original) The method of claim 21 wherein said loading even numbered extrema from a set of set processing elements into a second plurality of registers comprises:

loading said least significant byte of an extrema having an even position into a first register; transferring said least significant byte of said extrema in said first register into a second register and loading said most significant byte of said extrema into said first register;

transferring said most significant byte of said extrema in said first register into a third register and loading said least significant byte of another extrema having an even position into said first register; and

transferring said least significant byte of said another extrema in said first register into a fourth register and loading said most significant byte of said another extrema into said first register.

24. (original) The method of claim 22 wherein said comparing certain of said loaded odd numbered extrema to produce an odd extrema comprises:

comparing said least significant byte of said extrema in said second register to said least significant byte of said another extrema in said fourth register; and

comparing said most significant byte of said extrema in said third register to said most significant byte of said another extrema in said first register.

25. (original) The method of claim 23 wherein said comparing certain of said loaded even numbered extrema to produce an even extrema comprises:

comparing said least significant byte of said extrema in said second register to said least significant byte of said another extrema in said fourth register; and

comparing said most significant byte of said extrema in said third register to said most significant byte of said another extrema in said first register.

26. (original) The method of claim 24 wherein said comparing certain of said loaded odd numbered extrema to produce an odd extrema further comprises:

updating said second and third registers with said odd extrema;

loading said least significant byte of a next extrema having an odd position into said first register;

transferring said least significant byte of said next extrema in said first register into a fourth register and loading said most significant byte of said next extrema into said first register; and

repeating said comparing steps, said updating step, and said loading and transferring steps for each remaining extrema in an odd position within said data stream.

27. (original) The method of claim 25 wherein said comparing certain of said loaded even numbered extrema to produce an even extrema further comprises:

updating said second and third registers with said even extrema; and

loading said least significant byte of a next extrema having an even position into said first register;

transferring said least significant byte of said next extrema in said first register into a fourth register and loading said most significant byte of said next extrema into said first register; and

repeating said comparing steps, said updating step, and said loading and transferring steps for each remaining extrema in an even position within said data stream.

28. (original) The method of claim 21 further comprising:

determining a next dimensional extrema for a next dimension of said n-dimensional array, wherein said next dimensional extrema is related to said dimensional extrema, said next dimensional extrema having a most significant byte and a least significant byte; and

repeating said determining a next dimensional extrema for each of said n-dimensions, wherein each of said next dimensional extrema is related to a dimensional extrema from a previously selected dimension.

29. (original) The method of claim 21 wherein said loading odd numbered extrema from a set of said processing elements in a first dimension into a first plurality of registers and said loading even numbered extrema from a set of set processing elements into a second plurality of registers comprises:

loading a burst of said least significant bytes of said odd and said even numbered extrema, wherein said odd numbered least significant bytes are loaded into said first plurality of registers and said even numbered least significant bytes are loaded into said second plurality of registers; and

loading a burst of said most significant bytes of said odd and said even numbered extrema, wherein said odd numbered most significant bytes are loaded into said first plurality of registers and said even numbered most significant bytes are loaded into said second plurality of registers.

30. (currently amended) A computer readable memory device carrying a set of instructions which, when executed, perform a method comprising:

determining within each of said processing elements a first dimensional extrema for a first dimension of said an n-dimensional array of processing elements, wherein said first dimensional extrema is related to one or more local extrema of a plurality of values stored in said processing elements in said first dimension and wherein said first dimensional extrema has a most significant byte and a least significant byte;

determining within each of said processing elements a next dimensional extrema for a next dimension of said n-dimensional array, wherein said next dimensional extrema is related to one or more of said first dimensional extrema and wherein said next dimensional extrema has a most significant byte and a least significant byte; and

repeating said determining within each of said processing elements a next dimensional extrema for each of said n-dimensions, wherein each of said next dimensional extrema is related to a dimensional extrema from a previously selected dimension, until a global extrema is found;
and

saving said global extrema.